

1AP7 Rec'd PCT/PTO 11 JUL 2006'

PTO/SB/21 (09-04)

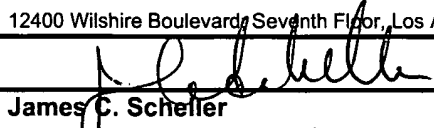
Approved for use through 07/31/2006. OMB 0651-0031

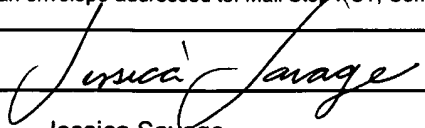
U.S. Patent and Trademark Office; U.S. DEPARTMENT OF COMMERCE

Under the Paperwork Reduction Act of 1995, no persons are required to respond to a collection of information unless it displays a valid OMB control number.

TRANSMITTAL FORM <small>(to be used for all correspondence after initial filing)</small>	Application Number	10/564,568	
	Filing Date	13 January 2006	
	First Named Inventor	Sergey Nikolaevich Zheltov	
	Art Unit	Not yet assigned	
	Examiner Name	Not yet assigned	
Total Number of Pages in This Submission	33	Attorney Docket Number	42390P16120

ENCLOSURES (Check all that apply)		
<input type="checkbox"/> Fee Transmittal Form <input type="checkbox"/> Fee Attached <input type="checkbox"/> Amendment/Reply <input type="checkbox"/> After Final <input type="checkbox"/> Affidavits/declaration(s) <input type="checkbox"/> Extension of Time Request <input type="checkbox"/> Express Abandonment Request <input type="checkbox"/> Information Disclosure Statement <input checked="" type="checkbox"/> Certified Copy of Priority Document(s) <input type="checkbox"/> Reply to Missing Parts/Incomplete Application <input type="checkbox"/> Reply to Missing Parts under 37 CFR 1.52 or 1.53	<input type="checkbox"/> Drawing(s) <input type="checkbox"/> Licensing-related Papers <input type="checkbox"/> Petition <input type="checkbox"/> Petition to Convert to a Provisional Application <input type="checkbox"/> Power of Attorney, Revocation <input type="checkbox"/> Change of Correspondence Address <input type="checkbox"/> Terminal Disclaimer <input type="checkbox"/> Request for Refund <input type="checkbox"/> CD, Number of CD(s) _____ <input type="checkbox"/> Landscape Table on CD	<input type="checkbox"/> After Allowance Communication to TC <input type="checkbox"/> Appeal Communication to Board of Appeals and Interferences <input type="checkbox"/> Appeal Communication to TC (Appeal Notice, Brief, Reply Brief) <input type="checkbox"/> Proprietary Information <input type="checkbox"/> Status Letter <input checked="" type="checkbox"/> Other Enclosure(s) (please identify below): Return Receipt Postcard
Remarks _____ Express Mail No. EV 841 071 808 US		

SIGNATURE OF APPLICANT, ATTORNEY, OR AGENT			
Firm Name	BLAKELY, SOKOLOFF, TAYLOR & ZAFMAN, LLP 12400 Wilshire Boulevard, Seventh Floor, Los Angeles, CA 90025-1030		
Signature			
Printed name	James C. Scheller		
Date	7/11/2006	Reg. No.	31,195

CERTIFICATE OF MAILING			
I hereby certify that this correspondence is being deposited with the United States Postal Service on the date shown below with sufficient postage as first class mail in an envelope addressed to: Mail Stop PCT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.			
Signature			
Typed or printed name	Jessica Savage	Date	7/11/2006

This collection of information is required by 37 CFR 1.5. The information is required to obtain or retain a benefit by the public which is to file (and by the USPTO to process) an application. Confidentiality is governed by 35 U.S.C. 122 and 37 CFR 1.11 and 1.14. This collection is estimated to 2 hours to complete, including gathering, preparing, and submitting the completed application form to the USPTO. Time will vary depending upon the individual case. Any comments on the amount of time you require to complete this form and/or suggestions for reducing this burden, should be sent to the Chief Information Officer, U.S. Patent and Trademark Office, U.S. Department of Commerce, P.O. Box 1450, Alexandria, VA 22313-1450. DO NOT SEND FEES OR COMPLETED FORMS TO THIS ADDRESS. SEND TO: MS PCT, Commissioner for Patents, P.O. Box 1450, Alexandria, VA 22313-1450.

If you need assistance in completing the form, call 1-800-PTO-9199 and select option 2.

РОСПАТЕНТ

Федеральное государственное учреждение



«Федеральный институт

промышленной собственности

Федеральной службы по интеллектуальной
собственности, патентам и товарным знакам»

(ФГУ ФИПС)

Бережковская наб., 30, корп. 1, Москва, Г-59, ГСП-5, 123995

Телефон 240- 60- 15. Телекс 114818 ПДЧ. Факс 234- 30- 58

Наш № 20/12-260

«11» мая 2006 г.

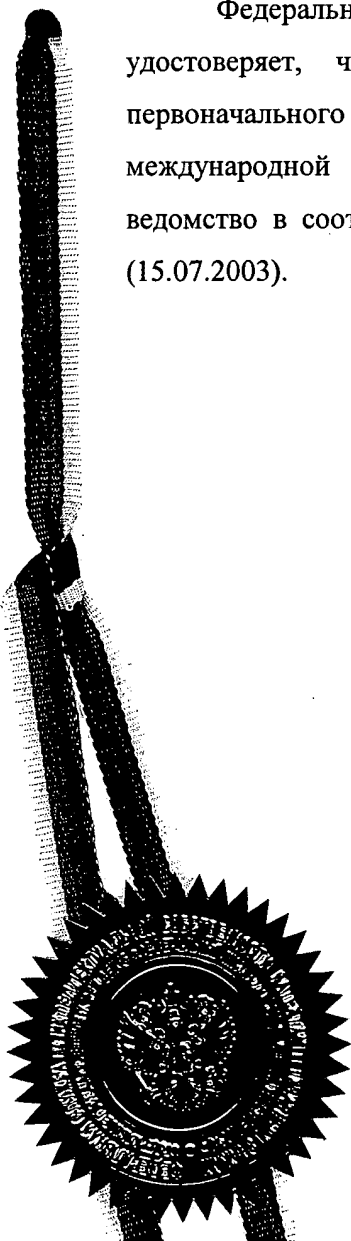
СПРАВКА

Федеральный институт промышленной собственности (далее - Институт) настоящим удостоверяет, что приложенные материалы являются точным воспроизведением первоначального заявления, описания, формулы, реферата и чертежей (если имеются) международной заявки № PCT/RU2003/00306, поданной в Институт как в Получающее ведомство в соответствии с Договором о патентной кооперации 15 июля 2003 года (15.07.2003).

**CERTIFIED COPY OF
PRIORITY DOCUMENT**

И.О. заведующего отделом 20

Т.Ф.Владимирова



РСТ

ЗАЯВЛЕНИЕ

Нижеподписавшийся
просит рассматривать настоящую международную
заявку в соответствии с Договором о патентной
кооперации

Заполняется получающим ведомством

РСТ/RU 0 3 / 0 0 3 0 6

Номер международной заявки

15 ИЮЛЯ 2003 (15.07.2003)

Дата международной подачи

RO/RU

Наименование получающего ведомства и штамп
«Международная заявка РСТ»
PCT INTERNATIONAL APPLICATION

№ дела заявителя или агента

(по желанию) (максимум 12 знаков) 2428

Графа I НАЗВАНИЕ ИЗОБРЕТЕНИЯ "A method of efficient performance
monitoring for symmetric multi-threading systems"

Графа II ЗАЯВИТЕЛЬ ☐ Данное лицо является также изобретателем

Имя и адрес: (Фамилия указывается перед именем, для юридического лица - полное уставное наименова-
ние. Адрес должен включать почтовый индекс и название страны. Если государство местожительства
низу не будет указано, то таковым будет считаться страна указанного в данной графе адреса)

INTEL, ZAKRYTOE AKTSIONERNOE OBSHCHESTVO

RU, 125252, Moscow, Sokol-10 Business Center,
Chapaevsky per., 14

Телефон №

Телефакс №

RO/RU

Телепринтер №

Регистрационный №
заявителя в Ведомстве

Государство (т.е. страна) гражданства:

Государство (т.е. страна) местожительства:

Данное лицо является
заявителем для:

всех указанных
государстввсех указанных
государств, кроме США

только США

государств, указанных в
дополнительной графе

RO/RU

Графа III ДРУГИЕ ЗАЯВИТЕЛИ И/ИЛИ (ДРУГИЕ) ИЗОБРЕТАТЕЛИ

Имя и адрес: (Фамилия указывается перед именем, для юридического лица - полное уставное наименова-
ние. Адрес должен включать почтовый индекс и название страны. Если государство местожительства
низу не будет указано, то таковым будет считаться страна указанного в данной графе адреса)

ZHELTON Sergei Nikolaevich
RU, 603950, Nizhny Novgorod, Turgenev st., 30

Данное лицо является:



только заявителем:



заявителем и изобретателем

только изобретателем (если отмечен
этот бокс, то ниже заполнять
не требуется)

Регистрационный №
заявителя в Ведомстве

Государство (т.е. страна) гражданства:

RU

Государство (т.е. страна) местожительства:

RU

Данное лицо является
заявителем для:

всех указанных
государстввсех указанных
государств, кроме США

только США

государств, указанных в
дополнительной графе

Другие заявители и/или (другие) изобретатели названы на листе продолжения

Графа IV АГЕНТ ИЛИ ОБЩИЙ ПРЕДСТАВИТЕЛЬ; ИЛИ АДРЕС ДЛЯ ПЕРЕПИСКИ

Указанное ниже лицо настоящим назначается (назначено) представлять
интересы заявителя(ей) в компетентных международных органах в качестве:



агента

общего
представителя

Имя и адрес: (Фамилия указывается перед именем, для юридического лица - полное уставное
наименование. Адрес должен включать почтовый индекс и название страны)

OBSHCHESTVO S OGRANICHENNOI
OTVETSTVENNOSTJU "SOJUZPATENT"
RU, 103735, Moscow, ul. Ilinka, d.5/2

Телефон №

925-16-61

Телефакс №

924-95-40

Телепринтер №

Регистрационный №
агента в Ведомстве



Адрес для переписки: Пометить этот бокс, если агент или общий представитель не назначаются (не назначены), а
указанный выше адрес используется только как специальный адрес для переписки

Графа III ДРУГИЕ ЗАЯВИТЕЛИ И/ИЛИ (ДРУГИЕ) ИЗОБРЕТАТЕЛИ*Если ни одна из следующих подграф не используется, этот лист не включается в заявление**Имя и адрес: (Фамилия указывается перед именем, для юридического лица - полное уставное наименование. Адрес должен включать почтовый индекс и название страны. Если государство местожительства внизу не будет указано, то таковым будет считаться страна указанного в данной графе адреса)*BRATANOV Stanislav Viktorovich
RU, 603950, Nizhny Novgorod,
Turgenev st., 30

Данное лицо является:

- ☐ только заявителем:
- ☒ заявителем и изобретателем
- ☐ только изобретателем (если отмечен этот бокс, то ниже заполнять не требуется)

Регистрационный №
заявителя в Ведомстве

Государство (т.е. страна) гражданства:

RU

Государство (т.е. страна) местожительства:

RU

Данное лицо является
заявителем для:☐всех указанных
государств☐всех указанных
государств, кроме США☒

только США

☐государств, указанных в
дополнительной графе*Имя и адрес: (Фамилия указывается перед именем, для юридического лица - полное уставное наименование. Адрес должен включать почтовый индекс и название страны. Если государство местожительства внизу не будет указано, то таковым будет считаться страна указанного в данной графе адреса)*BELENOV Roman Alexeevich
RU, 603950, Nizhny Novgorod,
Turgenev st., 30

Данное лицо является:

- ☐ только заявителем:
- ☒ заявителем и изобретателем
- ☐ только изобретателем (если отмечен этот бокс, то ниже заполнять не требуется)

Регистрационный №
заявителя в Ведомстве

Государство (т.е. страна) гражданства:

RU

Государство (т.е. страна) местожительства:

RU

Данное лицо является
заявителем для:☐всех указанных
государств☐всех указанных
государств, кроме США☒

только США

☐государств, указанных в
дополнительной графе*Имя и адрес: (Фамилия указывается перед именем, для юридического лица - полное уставное наименование. Адрес должен включать почтовый индекс и название страны. Если государство местожительства внизу не будет указано, то таковым будет считаться страна указанного в данной графе адреса)*KNYAZEV Alexandr Nikolaevich
RU, 603950, Nizhny Novgorod,
Turgenev st., 30

Данное лицо является:

- ☐ только заявителем:
- ☒ заявителем и изобретателем
- ☐ только изобретателем (если отмечен этот бокс, то ниже заполнять не требуется)

Регистрационный №
заявителя в Ведомстве

Государство (т.е. страна) гражданства:

RU

Государство (т.е. страна) местожительства:

RU

Данное лицо является
заявителем для:☐всех указанных
государств☐всех указанных
государств, кроме США☒

только США

☐государств, указанных в
дополнительной графе*Имя и адрес: (Фамилия указывается перед именем, для юридического лица - полное уставное наименование. Адрес должен включать почтовый индекс и название страны. Если государство местожительства внизу не будет указано, то таковым будет считаться страна указанного в данной графе адреса)*

Данное лицо является:

- ☐ только заявителем:
- ☐ заявителем и изобретателем
- ☐ только изобретателем (если отмечен этот бокс, то ниже заполнять не требуется)

Регистрационный №
заявителя в Ведомстве

Государство (т.е. страна) гражданства:

Государство (т.е. страна) местожительства:

Данное лицо является
заявителем для:☐всех указанных
государств☐всех указанных
государств, кроме США☐

только США

☐государств, указанных в
дополнительной графе☐

Другие заявители и/или (другие) изобретатели названы на другом листе для продолжения

Графа V УКАЗАНИЕ ГОСУДАРСТВ Пометьте нужные боксы ниже, должен быть отмечен как минимум один бокс

Настоящим делаются следующие указания в соответствии с правилом 4.9(a):

Региональный патент

- ☒ **AP** Патент ARIPO: GH Гана, GM Гамбия, KE Кения, LS Лесото, MW Малави, MZ Мозамбик, SD Судан, SL Сьерра-Леоне, SZ Свазиленд, TZ Объединенная Республика Танзания, UG Уганда, ZH Замбия, ZW Зимбабве, а также любое другое государство, являющееся Договаривающимся государством Протокола Хараре и PCT (если испрашивается иной вид охраны или статус, написать на пунктирной линии):
- ☒ **EA** Евразийский патент: AM Армения, AZ Азербайджан, BY Беларусь, KG Кыргызстан, KZ Казахстан, MD Республика Молдова, RU Российская Федерация, TJ Таджикистан, TM Туркменистан, а также любое другое государство, являющееся Договаривающимся государством Евразийской патентной конвенции и PCT
- ☒ **EP** Европейский патент: AT Австрия, BE Бельгия, CH и LI Швейцария и Лихтенштейн, CY Кипр, DE Германия, DK Дания, ES Испания, FI Финляндия, FR Франция, GB Великобритания, GR Греция, IE Ирландия, IT Италия, LU Люксембург, MC Монако, NL Нидерланды, PT Португалия, SE Швеция, TR Турция, а также любое другое государство, являющееся Договаривающимся государством Европейской патентной конвенции и PCT, ~~SI Словения~~
- ☒ **OA** Патент OAPI: BF Буркина Фасо, BJ Бенин, CF Центральная Африканская республика, CG Конго, CI Кот д'Ивуар, CM Камерун, GA Габон, GN Гвинея, GQ Экваториальная Гвинея, GW Гвинея-Бисау, ML Мали, MR Мавритания, NE Нигер, SN Сенегал, TD Чад, TG Того а также любое другое государство, являющееся членом OAPI и Договаривающимся государством PCT (если испрашивается иной вид охраны или статус, написать на пунктирной линии):

Национальный патент (если испрашивается иной вид охраны или статус, написать на пунктирной линии):

- | | | |
|---|---|--|
| <input checked="" type="checkbox"/> AE Объединенные Арабские Эмираты | <input checked="" type="checkbox"/> GM Гамбия | <input checked="" type="checkbox"/> OM Оман |
| <input checked="" type="checkbox"/> AG Антигуа и Барбуда | <input checked="" type="checkbox"/> HR Хорватия | <input checked="" type="checkbox"/> NZ Новая Зеландия |
| <input checked="" type="checkbox"/> AL Албания | <input checked="" type="checkbox"/> HU Венгрия | <input checked="" type="checkbox"/> PH Филиппины |
| <input checked="" type="checkbox"/> AM Армения | <input checked="" type="checkbox"/> ID Индонезия | <input checked="" type="checkbox"/> PL Польша |
| <input checked="" type="checkbox"/> AT Австрия | <input checked="" type="checkbox"/> IL Израиль | <input checked="" type="checkbox"/> PT Португалия |
| <input checked="" type="checkbox"/> AU Австралия | <input checked="" type="checkbox"/> IN Индия | <input checked="" type="checkbox"/> RO Румыния |
| <input checked="" type="checkbox"/> AZ Азербайджан | <input checked="" type="checkbox"/> IS Исландия | <input checked="" type="checkbox"/> RU Российская Федерация |
| <input checked="" type="checkbox"/> BA Босния и Герцеговина | <input checked="" type="checkbox"/> JP Япония | |
| <input checked="" type="checkbox"/> BB Барбадос | <input checked="" type="checkbox"/> KE Кения | <input checked="" type="checkbox"/> SD Судан |
| <input checked="" type="checkbox"/> BG Болгария | <input checked="" type="checkbox"/> KG Кыргызстан | <input checked="" type="checkbox"/> SE Швеция |
| <input checked="" type="checkbox"/> BR Бразилия | <input checked="" type="checkbox"/> KP Корейская народно-демократическая республика | <input checked="" type="checkbox"/> SG Сингапур |
| <input checked="" type="checkbox"/> BY Беларусь | <input checked="" type="checkbox"/> KR Республика Корея | <input checked="" type="checkbox"/> SI Словения |
| <input checked="" type="checkbox"/> BZ Белиз | <input checked="" type="checkbox"/> KZ Казахстан | <input checked="" type="checkbox"/> SK Словакия |
| <input checked="" type="checkbox"/> CA Канада | <input checked="" type="checkbox"/> LC Сент-Люсия | <input checked="" type="checkbox"/> SL Сьерра-Леоне |
| <input checked="" type="checkbox"/> CH and LI Швейцария и Лихтенштейн | <input checked="" type="checkbox"/> LK Шри Ланка | <input checked="" type="checkbox"/> TJ Таджикистан |
| <input checked="" type="checkbox"/> CN Китай | <input checked="" type="checkbox"/> LR Либерия | <input checked="" type="checkbox"/> TM Туркменистан |
| <input checked="" type="checkbox"/> CO Колумбия | <input checked="" type="checkbox"/> LS Лесото | <input checked="" type="checkbox"/> TN Тунис |
| <input checked="" type="checkbox"/> CR Коста Рика | <input checked="" type="checkbox"/> LT Литва | <input checked="" type="checkbox"/> TR Турция |
| <input checked="" type="checkbox"/> CU Куба | <input checked="" type="checkbox"/> LU Люксембург | <input checked="" type="checkbox"/> TT Тринидад и Тобаго |
| <input checked="" type="checkbox"/> CZ Чешская республика | <input checked="" type="checkbox"/> LV Латвия | <input checked="" type="checkbox"/> TZ Танзания |
| <input checked="" type="checkbox"/> DE Германия | <input checked="" type="checkbox"/> MA Марокко | <input checked="" type="checkbox"/> UA Украина |
| <input checked="" type="checkbox"/> DK Дания | <input checked="" type="checkbox"/> MD Республика Молдова | <input checked="" type="checkbox"/> UG Уганда |
| <input checked="" type="checkbox"/> DM Доминика | <input checked="" type="checkbox"/> MG Мадагаскар | <input checked="" type="checkbox"/> US Соединенные Штаты Америки |
| <input checked="" type="checkbox"/> DZ Алжир | <input checked="" type="checkbox"/> MK Бывшая Югославская республика Македония | <input checked="" type="checkbox"/> UZ Узбекистан |
| <input checked="" type="checkbox"/> EC Эквадор | <input checked="" type="checkbox"/> MN Монголия | <input checked="" type="checkbox"/> VN Вьетнам |
| <input checked="" type="checkbox"/> EE Эстония | <input checked="" type="checkbox"/> MW Малави | <input checked="" type="checkbox"/> YU Югославия |
| <input checked="" type="checkbox"/> ES Испания | <input checked="" type="checkbox"/> MX Мексика | <input checked="" type="checkbox"/> ZA Южная Африка |
| <input checked="" type="checkbox"/> FI Финляндия | <input checked="" type="checkbox"/> MZ Мозамбик | <input checked="" type="checkbox"/> ZH Замбия |
| <input checked="" type="checkbox"/> GB Великобритания | <input checked="" type="checkbox"/> NO Норвегия | <input checked="" type="checkbox"/> ZW Зимбабве |
| <input checked="" type="checkbox"/> GD Гренада | | |
| <input checked="" type="checkbox"/> GE Грузия | | |
| <input checked="" type="checkbox"/> GH Гана | | |

Боксы, зарезервированные для указания государств, которые стали участниками PCT после выпуска данного листа

<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>
<input type="checkbox"/>	<input type="checkbox"/>	<input type="checkbox"/>

Упоминание о предварительных указаниях: В дополнение к указаниям, сделанным выше, заявитель, в соответствии с правилом 4.9(b), делает также все указания, допустимые в соответствии с PCT, за исключением указания (указаний), приведенного в Дополнительной графе в качестве исключенных из данного упоминания, и заявляет, что эти дополнительные указания подлежат подтверждению, и что любое указание, не подтвержденное до истечения 15 месяцев с даты приоритета, должно считаться изъятым заявителем на момент истечения этого срока. (Подтверждение (включая оплату пошлины) должно быть представлено в получающее ведомство в пределах 15-месячного срока)

Графа VI ЗАЯВЛЕНИЕ НА ПРИОРИТЕТ

Настоящим заявляется приоритет следующей предшествующей заявки(ок) :

Дата подачи предшествующей заявки (день/месяц/год)	Номер предшествующей заявки	Если предшествующая заявка является:		
		национальной заявкой: страна	региональной заявкой: региональное ведомство	международной заявкой: получающее ведомство
(1)				
(2)				
(3)				
(4)				
(5)				

☐ Последующие заявления на приоритет указаны в Дополнительной графе

Получающему ведомству поручается подготовить и направить в Международное бюро заверенную копию предшествующей заявки(заявок)(только в том случае, если предшествующая заявка(заявки) была подана в ведомство, которое для настоящей международной заявки является получающим ведомством), указанную выше как:

☐ все ☐ (1) ☐ (2) ☐ (3) ☐ (4) ☐ (5) ☐ другое, см. Дополнительную графу

*Если предшествующей заявкой является заявка ARIPO, то должна быть указана, по крайней мере, одна страна-участница Парижской конвенции по охране промышленной собственности или одна страна-член Всемирной Торговой Организации, в которую была подана ранняя заявка (правило 4.10(b)(ii)).....

Графа VII МЕЖДУНАРОДНЫЙ ПОИСКОВЫЙ ОРГАН

Выбор международного поискового органа (ISA) (если компетентными в проведении международного поиска являются два или более международных поисковых органа, указать выбранный поисковый орган; можно использовать двубуквенный код):

ISA / RU

Просьба об использовании результатов ранее проведенного поиска; ссылка на такой поиск (если поиск был уже проведен или запрошен у Международного поискового органа ранее):

Дата (день/месяц/год)

Номер

Страна (или региональное ведомство)

Графа VIII ДЕКЛАРАЦИИ

Данное заявление содержит следующие декларации (ниже отметить необходимые боксы и указать в правой колонке количество каждого типа деклараций):

Количество деклараций

- ☐ Графа VIII (i) Декларация об удостоверении личности изобретателя
- ☐ Графа VIII (ii) Декларация о правомочности заявителя на дату международной подачи подавать заявку и получать патент
- ☐ Графа VIII (iii) Декларация о правомочности заявителя на дату международной подачи на заявление о приоритете в случае, если он не является заявителем, подавшим предшествующую заявку
- ☐ Графа VIII (iv) Декларация об авторстве на изобретение для целей указания Соединенных Штатов Америки
- ☐ Графа VIII (v) Декларация о не наносящих ущерб раскрытиях или изъятиях из-за отсутствия новизны

Графа IX КОНТРОЛЬНЫЙ ПЕРЕЧЕНЬ; ЯЗЫК ПОДАЧИ

Настоящая международная заявка содержит:

- (а) следующее количество листов на бумажном носителе:
- | | |
|--|----|
| заявление(включая декларацию) | 5 |
| описание (исключая перечень последовательностей) | 15 |
| формула | 7 |
| реферат | 1 |
| чертежи | 3 |

Предварительное число листов : 31

часть описания с перечнем последовательностей (действительное число листов, представленных на бумажном носителе, независимо от представления в машиночитаемой форме; см. ниже пункт (b))

Общее число листов : 31

(b) перечень последовательностей представлен в машиночитаемой форме

- (i) ☐ только (в соответствии с разделом 801(a)(i))
- (ii) ☐ как приложение к представленному на бумажном носителе(в соответствии с разделом 801(a)(ii))

Тип и количество носителей (дискета, CD-ROM, CD-R или другое), на котором представлен перечень последовательностей (дополнительно к указанному в пункте 9(ii) в правой колонке):

К настоящей международной заявке приложены следующие документы (ниже следует отметить соответствующие боксы и указать с права количество приложений каждого вида):

- ☐ лист расчета пошлин
- ☐ оригинал отдельной доверенности
- ☐ оригинал генеральной доверенности
- ☐ копия генеральной доверенности; ссылка на номер, если имеется
- ☐ разъяснения по поводу отсутствия подписи
- ☐ приоритетный(ые) документ(ы), указанный в графе VI под №
- ☐ перевод международной заявки на (язык)
- ☐ информация о депонировании микроорганизмов или другого биологического материала
- ☐ перечень последовательностей в машиночитаемой форме(указать тип и число носителей (дискета, CD-ROM, CD-R или иное))
 - ☐ копия, представленная для целей международного поиска в соответствии с правилом 13 ter (и не являющаяся частью международной заявки)
 - ☐ (только в случае, если слева отмечены бокс(b)(i) или (b)(ii)) дополнительно представленная копия, если допустимо, копия для целей международного поиска в соответствии с правилом 13 ter
 - ☐ вместе с соответствующим представлением перечня последовательностей, как его заявление отмечено слева
- ☐ иное (указать)

Кол-во приложений

Фигура чертежей, предлагаемая для публикации с рефератом:

Язык подачи

международной заявки: GB

Графа X ПОДПИСЬ ЗАЯВИТЕЛЯ, АГЕНТА ИЛИ ОБЩЕГО ПРЕДСТАВИТЕЛЯ

Рядом с каждой подписью указать фамилию каждого подписавшего и указать, в каком качестве он подписал заявление (если это не очевидно из данных, приведенных в заявлении).

General Director
of OOO "Soyuzpatent"

Felitsyna S.B.

Заполняется получающим ведомством

1. Дата фактического получения международной заявки:	15 ИЮЛЯ 2003 (15.07.2003)	2. Чертежи: <input checked="" type="checkbox"/> получены: <input type="checkbox"/> не получены:
3. Исправленная дата при более позднем, но своевременном получении страниц или чертежей, доукомплектовывающих предполагаемую международную заявку:		
4. Дата своевременного получения требуемых исправлений согласно статье 11(2) PCT:		
5. Международный поисковый орган (если компетентны два и более): ISA/ RU	6. <input type="checkbox"/> Направление копии для поиска задержано впредь до уплаты пошлины за поиск	

Заполняется Международным бюро

Дата получения регистрационного экземпляра
Международным бюро:

**A Method of Efficient Performance Monitoring for
Symmetric Multi-Threading Systems**

5 A portion of the disclosure of this patent document contains material that is subject to copyright protection. The copyright owner has no objection to the facsimile reproduction by anyone of the patent document or the patent disclosure, as it appears in the Patent and Trademark Office patent file or records, but otherwise reserves all copyright rights whatsoever.

10 BACKGROUND

1. FIELD

15 The present invention relates generally to performance measurement techniques and, more specifically, to measurement of performance of an execution thread within a symmetric multi-threading (SMT) system.

2. DESCRIPTION

20 It is a general practice to increase the computational performance by organizing parallel program execution. There are a number of methods to achieve this, including, but not limited to, out-of-order instruction execution, multiple data operands, shared memory multi-processor systems, distributed computations, and so forth. One of the popular and relatively inexpensive approaches is to combine multiple execution cores within one physical processor, or even provide separate execution state containers and control logic to share multiple processing units of a physical processor. The latter statement is applicable, for example, to the Hyper-Threading technology commercially available from Intel Corporation, which provides better utilization of various execution units incorporated in a processor.

25 Measurement of a processor's (program's) performance is one of the main tasks to be solved when building an efficient computational system. For single processor systems, performance monitoring is a matter of correctly written software, given that the processor (or other hardware components) provides the necessary resources. The performance monitoring task may be more difficult for SMT systems: performance monitoring

30

hardware support may vary considerably, and the interaction between hardware and software parts of performance monitoring system becomes more complicated.

Possible difficulties that can arise include the lack of performance monitoring resources (e.g., performance counters) to monitor the activity of all processing units (e.g., logical threads or processors) within a physical package, and no hardware support of asynchronous and independent measurements performed on a per-thread (per-logical processor/unit) basis.

Therefore, a need exists for the capability to efficiently monitor the performance of multi-threading systems taking into account the possible lack of hardware resources.

BRIEF DESCRIPTION OF THE DRAWINGS

The features and advantages of the present invention will become apparent from the following detailed description of the present invention in which:

Figure 1 is a diagram illustrating the dedication of hardware resources to execution threads according to an embodiment of the present invention;

Figure 2 is a flow diagram illustrating the initiation of the performance monitoring process according to an embodiment of the present invention; and

Figure 3 is a flow diagram illustrating the completion of the performance monitoring operation according to an embodiment of the present invention.

DETAILED DESCRIPTION

Embodiments of the invention described herein may be applicable to performance monitoring conducted on an execution thread basis within a symmetric multi-threading (SMT) system. One embodiment of the present invention may be used in a system built on Intel Corporation's Hyper-Threading (HT) technology to enable effective performance monitoring on a per logical processor basis.

Reference in the specification to "one embodiment" or "an embodiment" of the present invention means that a particular feature, structure or characteristic described in connection with the embodiment is included in at least one embodiment of the present invention. Thus, the appearances of the phrase "in one embodiment" appearing in various places throughout the specification are not necessarily all referring to the same embodiment.

It is not always possible to provide independent hardware support for simultaneous monitoring of multiple execution threads or logical execution modules (e.g., logical processors for HT). Thus, many useful measurements can be performed for either all execution threads or for a specified subset, depending on a particular hardware implementation. Embodiments of the present invention relate to the case of limited performance monitoring resources and enable quasi-independent measurements for each execution thread or logical execution unit. That is, whenever a thread (logical unit) initiates measurements, the overall performance monitoring results are computed correctly, but the distribution of the results to any particular thread (logical unit) depends on a particular hardware implementation.

The following definitions may be useful in understanding embodiments of the present invention described herein.

A performance monitoring unit is a device (whether external, integrated, or a specific functional block within a primary unit) intended for measuring (monitoring) operational characteristics of a primary device (unit) or system.

An execution thread is a program to be executed by a processing unit (e.g., processor) independently and (if possible) concurrently with other programs, and the state of the processing unit (execution context) associated with such a program.

A logical execution unit is a specific processing unit that executes a program concurrently with other processing units, maintains a program execution state, and shares system resources with similar units within a primary processing unit.

One logical execution unit is supposed to run one execution thread (program) at a time. Therefore, for purposes of describing embodiments of the present invention there is no essential difference between the two terms. The methods described herein may be applicable to any processing system that may have performance monitoring resources shared between multiple processing units as well as multiple program threads as the latter are supported by hardware.

Hereinafter the term 'execution unit' denotes both an execution thread and a logical execution unit.

Figure 1 illustrates the structure of a performance monitoring unit (PMU) and three types of resource sharing that may occur in a symmetric multi-threading system. A PMU comprises counter logic 10, control logic 12, and execution unit indicator logic 14. In some embodiments, the execution unit indicator logic may be a part of the control logic. In

a system that supports multiple execution units (EUs) within a physical package and provides each EU with a separate PMU for any given performance monitoring functionality, all performance monitoring data may be collected independently and asynchronously on an EU's demand. There are, however, a number of systems with limited
5 PMU resources (e.g., Intel Corporation's Pentium4 processor with Hyper-Threading technology enabled) that need to be shared between multiple execution units. One of the examples of such sharing may be a system that has only one PMU that is capable of collecting performance data for either one or all execution units. The former case (one EU to be monitored) generally results in undercounting of performance data, while the latter
10 case will produce overcounted results. To handle both cases, a system that implements the present invention needs to emulate the execution unit indicators 16 for each EU by means of a request allocation as described below.

Most of the current state-of-the-art systems provide a capability to set up a PMU to collect performance data for a subset of execution units by furnishing additional EU-
15 indicators 18. Typically, the number of additional EU-indicators equals the number of execution units within a package, otherwise, if the number of EU-indicators appears to be less, the above described single EU-indicator conditions hold true for this case.

The performance monitoring process is illustrated in Figures 2 and 3.

According to embodiments of the present invention, the performance monitoring is
20 started or stopped upon a request from an execution unit. A system implementing the present method should be capable of maintaining the correct sequence of such requests, insuring that a stop request always follows a start request or establishing the start/stop correspondence in any other applicable manner, e.g., providing a nested request support or ignoring excessive requests. As the requests may appear simultaneously, a special
25 arbitration step 20 may be used to guarantee the exclusive use of a PMU. Once exclusive execution is acquired, the start request is allocated at block 22, that is, a special table (provided for this purpose) or PMU (if supported by hardware) field may be filled with a value indicating that a request to start performance monitoring operation is pending for a specific execution unit. If there is only one request currently allocated, the PMU may be
30 programmed at block 24 to start collecting performance monitoring data for the EU that allocated the request. At block 26, the PMU counter may be set to a predefined value if the hardware supports counter initialization; otherwise, the current counter value may be stored in a special memory area as an initial value and may be later subtracted from a final

value when the performance monitoring operation stops. In case there is more than one request already allocated, the requesting EU may be added to the set of EUs the PMU currently collects performance data for if such a possibility is supported by the PMU's hardware (PMU has a free EU-indicator).

5 Thus, performance monitoring process starts, and one counter of one PMU accumulates performance data for all execution units as they request this operation.

To stop the operation for an EU, a stop request may be issued by this execution unit. The arbitration may be performed at block 100 to acquire exclusive processing of each stop request. Then, at block 102, the stop request may be removed from the special
10 table (see above) or PMU EU-indicator field (if supported by hardware). If there are no more requests allocated, the PMU may be programmed to stop collecting performance data at block 104. The final performance value may then be obtained at block 106. If there are requests from other EUs, active or pending, and the current EU belongs to the set of EUs the PMU collects data for (i.e., the request is active), the PMU may be programmed to stop
15 collecting data for the current EU (if such a possibility is supported by the PMU's hardware). Then, one skilled in the art will recognize the option, based on the knowledge of a particular system architecture and hardware performance monitoring capabilities, of retrieving the final performance value at block 110, setting the initial value equal to the value retrieved or reprogramming the performance monitoring unit to start counting from a predefined value if the retrieving and reprogramming procedures do not substantially affect
20 performance monitoring results. Then, another EU needs to be selected at block 112 in order to be added to the set of EUs to accumulate data for at block 114. In case the current request is not within the set of active EUs (pending request, emulated by the EU-indicator 16), such a request may be discarded, and a zero or any predefined value may be returned
25 as the performance monitoring result.

Thus, one embodiment of the present invention may be a system that collects performance monitoring data in one PMU counter for all execution units, and returns the performance monitoring results either each time all the EUs complete their operation, or each time a EU that happens to fall within a set of active EUs requests for completion. This
30 means that the data collected pertains to all EUs and the total value is computed correctly (except for the described above cases of no hardware support for EU indicators) but the distribution of the final values to the EUs is considered system dependent. Still, even this implementation dependent information on the performance data distribution may be useful,

because it reflects the real-time EU interaction features and may be useful for many other types of system performance analysis.

For an exemplary embodiment of the present invention implemented in Assembler language refer to Appendix A. The Assembler code is provided for the purpose of illustration only and does not constitute a complete software performance monitoring system. Furthermore, one skilled in the art will recognize that embodiments of the present invention may be implemented in other ways and using other programming languages.

The techniques described herein are not limited to any particular hardware or software configuration; they may find applicability in any computing or processing environment. The techniques may be implemented in logic embodied in hardware, software, or firmware components, or a combination of the above. The techniques may be implemented in programs executing on programmable machines such as mobile or stationary computers, personal digital assistants, set top boxes, cellular telephones and pagers, and other electronic devices, that each include a processor, a storage medium readable by the processor (including volatile and non-volatile memory and/or storage elements), at least one input device, and one or more output devices. Program code is applied to the data entered using the input device to perform the functions described and to generate output information. The output information may be applied to one or more output devices. One of ordinary skill in the art may appreciate that the invention can be practiced with various computer system configurations, including multiprocessor systems, minicomputers, mainframe computers, and the like. The invention can also be practiced in distributed computing environments where tasks may be performed by remote processing devices that are linked through a communications network.

Each program may be implemented in a high level procedural or object oriented programming language to communicate with a processing system. However, programs may be implemented in assembly or machine language, if desired. In any case, the language may be compiled or interpreted.

Program instructions may be used to cause a general-purpose or special-purpose processing system that is programmed with the instructions to perform the operations described herein. Alternatively, the operations may be performed by specific hardware components that contain hardwired logic for performing the operations, or by any combination of programmed computer components and custom hardware components. The methods described herein may be provided as a computer program product that may

include a machine readable medium having stored thereon instructions that may be used to program a processing system or other electronic device to perform the methods. The term "machine readable medium" used herein shall include any medium that is capable of storing or encoding a sequence of instructions for execution by the machine and that cause the machine to perform any one of the methods described herein. The term "machine readable medium" shall accordingly include, but not be limited to, solid-state memories, optical and magnetic disks, and a carrier wave that encodes a data signal. Furthermore, it is common in the art to speak of software, in one form or another (e.g., program, procedure, process, application, module, logic, and so on) as taking an action or causing a result. Such expressions are merely a shorthand way of stating the execution of the software by a processing system to cause the processor to perform an action or produce a result.

While this invention has been described with reference to illustrative embodiments, this description is not intended to be construed in a limiting sense. Various modifications of the illustrative embodiments, as well as other embodiments of the invention, which are apparent to persons skilled in the art to which the invention pertains are deemed to lie within the spirit and scope of the invention.

APPENDIX A

© 2002 Intel Corporation

5 A code example to count the number of bus accesses from a Pentium4 processor with Hyper-Threading technology enabled.

```

    ;;; a function to perform arbitration
    syncHT proc    near
        ;;; IN bh == Local APIC ID
        ;;; OUT eax -> spin lock flag
10      movzx  eax,bh
        shr   eax,1
        lea   eax,[pml_sync_HT + eax]
        call  acquire_spin_lock
15      ret
    syncHT endp

    ;;; a function to start counting
    busproc_restart proc    near
20      mov    eax,1
        cpuid
        shr   ebx,16
        or    bl,bl    ;;; no HT when zero
        jz    no_HT
25      call   syncHT
        push  eax
    ;;; read ESCR
        xor   eax,eax
        xor   edx,edx
30      mov    ecx,msr_fsb_escr0
        rdmsr
        test  bh,01h
        jnz   cpu1

```

```
;;; executing on logical CPU0
```

```
cpu0:
```

```
;;; if Tx clear, program own CCCR to start counting
```

```

5      ;;; eax[3..2] == T0
      ;;; eax[1..0] == T1
      test    eax,03h
      jnz     T1set
      mov     ecx,msr_fsb_escr0
10     mov     eax,busproc_escr_mask2 OR busproc_escr_T0
      wrmsr
      ;;; clear the counter
      mov     eax,pml_initial_count
      mov     edx,pml_initial_count + 4
15     and     edx,0ffh      ;;; 40-bit counters
      mov     ecx,msr_bpu_counter0
      wrmsr
      mov     ecx,msr_bpu_cccr0
      mov     eax,busproc_cccr_mask_PMI0
20     wrmsr
      jmp     HT_exit

```

```
;;; else set T-own in ESCR
```

```
T1set:
```

```

25     mov     ecx,msr_fsb_escr0
      or      eax,busproc_escr_mask2 OR busproc_escr_T0
      wrmsr
      jmp     HT_exit

```

```
30    ;;; executing on logical CPU1
```

```
cpu1:
```

```
;;; if Tx clear, program own CCCR to start counting
```

```
      ;;; eax[3..2] == T0
```

10

```

    ;;; eax[1..0] == T1
    test    eax,0ch
    jnz     T0set
    mov     ecx,msr_fsb_escal0
5    mov     eax,busproc_escr_mask2 OR busproc_escr_T1
    wrmsr
    ;;; clear the counter
    mov     eax,pml_initial_count
    mov     edx,pml_initial_count + 4
10    and     edx,0ffh    ;;; 40-bit counters
    mov     ecx,msr_bpu_counter1
    wrmsr
    mov     ecx,msr_bpu_cccr1
    mov     eax,busproc_cccr_mask_PMI1
15    wrmsr
    jmp     HT_exit

```

```

    ;;; else set T-own in ESCR

```

```

T0set:

```

```

20    mov     ecx,msr_fsb_escal0
    or      eax,busproc_escr_mask2 OR busproc_escr_T1
    wrmsr

```

```

HT_exit:

```

```

    pop     eax
25    call    release_spin_lock
    ret

```

```

no_HT:

```

```

    mov     eax,pml_initial_count
    mov     edx,pml_initial_count + 4
30    and     edx,0ffh    ;;; 40-bit counters
    mov     ecx,msr_bpu_counter0
    wrmsr
    mov     ecx,msr_fsb_escal0

```

11

```

    mov     eax,busproc_escr_mask2 OR busproc_escr_T0
    wrmsr
    mov     ecx,msr_bpu_cccr0
    mov     eax,busproc_cccr_mask_PMI0
5         wrmsr
    ret
busproc_restart endp

;;; a function to stop counting and retrieve final value
10 busproc_freeze_read  proc  near
    ;;; OUT edx:eax = current count
    mov     eax,1
    cpuid
    shr     ebx,16
15    or     bl,bl ;;; no HT when zero
    jz      no_HT
    call    syncHT
    push    eax
    ;;; read ESCR
20    xor     eax,eax
    xor     edx,edx
    mov     ecx,msr_fsb_escr0
    rdmsr
    test    bh,01h
25    jnz     cpu1

    ;;; executing on logical CPU0
cpu0:
    ;;; if Tx clear, program own CCCR to stop counting
30    ;;; eax[3..2] == T0
    ;;; eax[1..0] == T1
    test    eax,03h
    jnz     T1set

```

```
    ;;; stop counting
    mov    eax,busproc_cccr_stop_mask
    xor    edx,edx
    mov    ecx,msr_bpu_cccr0
5    wrmsr
    ;;; clear ESCR
    mov    ecx,msr_fsb_escr0
    xor    eax,eax
    xor    edx,edx
10   wrmsr
    ;;; read count into edx:eax
    mov    ecx,msr_bpu_counter0
    rdmsr
    jmp    HT_exit
15
    ;;; else
    T1set:
    ;;; clear T-own in ESCR
        and    eax,NOT busproc_escr_T0
20   mov    ecx,msr_fsb_escr0
        wrmsr
    ;;; read own CCCR
        mov    ecx,msr_bpu_cccr0
        rdmsr
25   test   eax,cccr_enabled
        jz     disabled0
    enabled0:
        ;;; program the other's CCCR
        mov    eax,pml_initial_count
30   mov    edx,pml_initial_count + 4
        and    edx,0ffh    ;;; 40-bit counters
        mov    ecx,msr_bpu_counter1
        wrmsr
```

13

```

    mov     ecx,msr_bpu_cccr1
    mov     eax,busproc_cccr_mask_PMI1
    wrmsr
    ;;; stop counting
5      mov     eax,busproc_cccr_stop_mask
    xor     edx,edx
    mov     ecx,msr_bpu_cccr0
    wrmsr
    ;;; read count into edx:eax
10     mov     ecx,msr_bpu_counter0
    rdmsr
    jmp     HT_exit

disabled0:
15     ;;; return zero count
    xor     edx,edx
    xor     eax,eax
    jmp     HT_exit

20     ;;; executing on logical CPU1
    cpu1:
    ;;; if Tx clear, program own CCCR to stop counting
    ;;; eax[3..2] == T0
    ;;; eax[1..0] == T1
25     test    eax,0ch
    jnz     T0set
    ;;; stop counting
    mov     eax,busproc_cccr_stop_mask
    xor     edx,edx
30     mov     ecx,msr_bpu_cccr1
    wrmsr
    ;;; clear ESCR
    mov     ecx,msr_fsb_escr0

```

```

    xor    eax,eax
    xor    edx,edx
    wrmsr
    ;;; read count into edx:eax
5      mov    ecx,msr_bpu_counter1
    rdmsr
    jmp    HT_exit

    ;;; else
10     T0set:
    ;;; clear T-own in ESCR
        and    eax,NOT busproc_escr_T1
        mov    ecx,msr_fsb_escr0
        wrmsr
15     ;;; read own CCCR
        mov    ecx,msr_bpu_cccr1
        rdmsr
        test   eax,cccr_enabled
        jz     disabled1
20     enabled1:
        ;;; program the other's CCCR
        mov    eax,pml_initial_count
        mov    edx,pml_initial_count + 4
        and    edx,0ffh    ;;; 40-bit counters
25     mov    ecx,msr_bpu_counter0
        wrmsr
        mov    ecx,msr_bpu_cccr0
        mov    eax,busproc_cccr_mask_PMI0
        wrmsr
30     ;;; stop counting
        mov    eax,busproc_cccr_stop_mask
        xor    edx,edx
        mov    ecx,msr_bpu_cccr1
```

15

```
wrmsr
;;; read count into edx:eax
mov ecx,msr_bpu_counter1
rdmsr
5 jmp HT_exit
```

disabled1:

```
;;; return zero count
xor edx,edx
10 xor eax,eax
```

HT_exit:

```
xchg eax,[esp]
call release_spin_lock
15 pop eax
ret
```

no_HT:

```
;;; stop counting
20 mov eax,busproc_cccr_stop_mask
xor edx,edx
mov ecx,msr_bpu_cccr0
wrmsr
;;; clear ESCR
25 mov ecx,msr_fsb_escr0
xor eax,eax
xor edx,edx
wrmsr
;;; read count into edx:eax
30 mov ecx,msr_bpu_counter0
rdmsr
ret
```

busproc_freeze_read endp

CLAIMS

What is claimed is:

1. In a system that shares performance monitoring units between multiple
5 execution units, a method comprising:

receiving a request from an execution unit to at least one of start and stop
performance monitoring operation;

maintaining a correct sequence of requests to start and stop performance monitoring
operation;

10 performing arbitration to acquire exclusive execution for one of a plurality of
request initiators;

allocating a request to start performance monitoring operation;

determining a number of allocated requests to start performance monitoring
operation;

15 initiating performance monitoring operation;

removing a request to start performance monitoring operation;

determining an active mode of performance monitoring operation; and

completing performance monitoring operation.

2. The method of claim 1, wherein maintaining the correct sequence of
20 requests comprises ensuring that the stop request follows the start request for the same
execution unit, and additional requests are ignored.

3. The method of claim 1, wherein allocating the request comprises indicating
to the system that a request to start performance monitoring operation is pending for a
specific execution unit.

25 4. The method of claim 1, wherein allocating the request and initiating of
performance monitoring operation are performed if a request to start performance
monitoring operation was received.

5. The method of claim 1, wherein removing the request comprises indicating
to the system that no request to start performance monitoring operation is pending for the
30 current execution unit.

6. The method of claim 1, wherein removing the request, determining active
mode, and completing of performance monitoring operation are performed if a request to
stop performance monitoring operation was received.

7. The method of claim 1, wherein initiating performance monitoring operation comprises programming the performance monitoring unit to start collecting performance data for the execution unit that requested said operation if no other request was previously allocated.

5 8. The method of claim 7, further comprising at least one of setting an initial performance value to the current value of performance monitoring unit counter and setting the initial performance value and the performance monitoring unit counter to a predefined value.

9. The method of claim 1, wherein initiating performance monitoring
10 operation further comprises programming the performance monitoring unit to start collecting performance data for the execution unit that requested said operation, in addition to collecting performance data of other execution units which previously requested said operation if there are other requests previously allocated.

10. The method of claim 1, wherein determining the active mode of
15 performance monitoring operation comprises detecting whether programming of performance monitoring unit was performed for the execution unit that requested to stop performance monitoring operation.

11. The method of claim 1, wherein completing performance monitoring
20 operation comprises retrieving final performance data and programming the performance monitoring unit to stop collecting of performance data if no other requests for the performance monitoring unit are allocated.

12. The method of claim 1, wherein completing performance monitoring
25 operation further comprises programming the performance monitoring unit to stop collecting of performance data for the execution unit that requested said operation, if there are other requests previously allocated and the performance monitoring unit was in active mode for the execution unit.

13. The method of claim 12, further comprising retrieving current
30 performance data, setting the initial performance value equal to the value retrieved or reprogramming the performance monitoring unit to start counting from a predefined value and setting the initial performance value equal to the predefined value.

14. The method of claim 12, further comprising:
selecting another execution unit;

programming the performance monitoring unit to start collecting performance data for the selected execution unit.

15. The method of claim 14, wherein selecting another execution unit comprises selecting, by external means, of a request previously allocated by another execution unit and determining the execution unit that allocated said request.

16. The method of claim 14, wherein programming the performance monitoring unit further comprises enabling performance data collection for the selected execution unit in addition to performance data of other execution units which previously requested said operation.

17. An article comprising: a machine accessible medium having a plurality of machine readable instructions, wherein when the instructions are executed by a processor, the instructions provide for sharing of performance monitoring units between multiple execution units by:

receiving a request from an execution unit to at least one of start and stop performance monitoring operation;

maintaining a correct sequence of requests to start and stop performance monitoring operation;

performing arbitration to acquire exclusive execution for one of a plurality of request initiators;

allocating a request to start performance monitoring operation;
determining a number of allocated requests to start performance monitoring operation;

initiating performance monitoring operation;

removing a request to start performance monitoring operation;

determining an active mode of performance monitoring operation; and
completing performance monitoring operation.

18. The article of claim 17, wherein instructions for maintaining the correct sequence of requests comprise instructions for ensuring that the stop request follows the start request for the same execution unit, and additional requests are ignored.

19. The article of claim 17, wherein instructions for allocating the request comprise instructions for indicating to the system that a request to start performance monitoring operation is pending for a specific execution unit.

20. The article of claim 17, wherein instructions for allocating the request and initiating of performance monitoring operation are executed if a request to start performance monitoring operation was received.

21. The article of claim 17, wherein instructions for removing the request
5 comprise instructions for indicating to the system that no request to start performance monitoring operation is pending for the current execution unit.

22. The article of claim 17, wherein instructions for removing the request, determining active mode, and completing of performance monitoring operation are executed if a request to stop performance monitoring operation was received.

10 23. The article of claim 17, wherein instructions for initiating performance monitoring operation comprise instructions for programming the performance monitoring unit to start collecting performance data for the execution unit that requested said operation if no other request was previously allocated.

24. The article of claim 23, further comprising instructions for at least one of
15 setting an initial performance value to the current value of a performance monitoring unit counter and setting the initial performance value and the performance monitoring unit counter to a predefined value.

25. The article of claim 17, wherein instructions for initiating of performance monitoring operation further comprise instructions for programming the performance
20 monitoring unit to start collecting performance data for the execution unit that requested said operation, in addition to performance data of other execution units which previously requested said operation if there are other requests previously allocated.

26. The article of claim 17, wherein instructions for determining the active mode of performance monitoring operation comprise instructions for detecting whether
25 programming of performance monitoring unit was performed for the execution unit that requested to stop performance monitoring operation.

27. The article of claim 17, wherein instructions for completing performance monitoring operation comprise instructions for retrieving final performance data and programming the performance monitoring unit to stop collecting of performance data if no
30 other requests for the performance monitoring unit are allocated.

28. The article of claim 17, wherein instructions for completing performance monitoring operation further comprise instructions for programming the performance monitoring unit to stop collecting of performance data for the execution unit that requested

said operation, if there are other requests previously allocated and the performance monitoring unit was in active mode for the current execution unit.

29. The article of claim 28, further comprising instructions for retrieving current performance data, setting the initial performance value equal to the value retrieved or reprogramming performance monitoring unit to start counting from a predefined value and setting the initial performance value equal to the predefined value.

30. The article of claim 28, further comprising instructions for:
selecting another execution unit;
programming the performance monitoring unit to start collecting of performance data for the selected execution unit.

31. The article of claim 30, wherein instructions for selecting another execution unit comprise instructions for selecting, by external means, of a request previously allocated by another execution unit and determining the execution unit that allocated said request.

32. The article of claim 30, wherein instructions for programming the performance monitoring unit further comprise instructions for enabling performance data collection for the selected execution unit in addition to performance data of other execution units which previously requested said operation.

33. A system that shares performance monitoring units between multiple execution units comprising:

logic to receive a request from an execution unit to at least one of start and stop performance monitoring operation;

logic to maintain a correct sequence of requests to start or stop performance monitoring operation;

logic to perform arbitration to acquire exclusive execution for one of a plurality of request initiators;

logic to allocate a request to start performance monitoring operation;

logic to determine a number of allocated requests to start performance monitoring operation;

logic to initiate performance monitoring operation;

logic to remove a request to start performance monitoring operation;

logic to determine an active mode of performance monitoring operation; and

logic to complete performance monitoring operation.

34. The system of claim 33, wherein logic to maintain a correct sequence of requests comprises logic to ensure that the stop request follows the start request for the same execution unit, and additional requests are ignored.

35. The system of claim 33, wherein logic to allocate the request comprises
5 logic to indicate to the system that a request to start performance monitoring operation is pending for a specific execution unit.

36. The system of claim 33, wherein logic to allocate the request and initiate performance monitoring operation is activated if a request to start performance monitoring operation was received.

10 37. The system of claim 33, wherein logic to remove the request comprises logic to indicate to the system that no request to start performance monitoring operation is pending for the current execution unit.

38. The system of claim 33, wherein logic to remove the request, determine active mode, and complete performance monitoring operation is activated if a request to
15 stop performance monitoring operation was received.

39. The system of claim 33, wherein logic to initiate performance monitoring operation comprises logic to program the performance monitoring unit to start collecting performance data for the execution unit that requested said operation if no other request was previously allocated.

20 40. The system of claim 39, further comprising logic to at least one of set an initial performance value to the current value of a performance monitoring unit counter and set the initial performance value and the performance monitoring unit counter to a predefined value.

41. The system of claim 33, wherein logic to initiate performance monitoring
25 operation further comprises logic to program the performance monitoring unit to start collecting performance data for the execution unit that requested said operation, in addition to performance data of other execution units which previously requested said operation if there are other requests previously allocated.

42. The system of claim 33, wherein logic to determine the active mode of
30 performance monitoring operation comprises logic to detect whether programming of performance monitoring unit was performed for the execution unit that requested to stop performance monitoring operation.

43. The system of claim 33, wherein logic to complete performance monitoring operation comprises logic to retrieve final performance data and to program the performance monitoring unit to stop collecting of performance data if no other requests for the performance monitoring unit are allocated.

5 44. The system of claim 33, wherein logic to complete performance monitoring operation further comprises logic to program the performance monitoring unit to stop collecting performance data for the execution unit that requested said operation, if there are other requests previously allocated and the performance monitoring unit was in the active mode for the current execution unit.

10 45. The system of claim 44, further comprising logic to retrieve current performance data, to set the initial performance value equal to the value retrieved or to reprogram performance monitoring unit to start counting from a predefined value and set the initial performance value equal to the predefined value.

46. The system of claim 44, further comprising:

15 logic to select another execution unit;
logic to program the performance monitoring unit to start collecting performance data for the selected execution unit.

20 47. The system of claim 46, wherein logic to select another execution unit comprises logic to select, by external means, a request previously allocated by another execution unit and to determine the execution unit that allocated said request.

48. The system of claim 46, wherein logic to program the performance monitoring unit further comprises logic to enable performance data collection for the selected execution unit in addition to performance data of other execution units which previously requested said operation.

ABSTRACT OF THE DISCLOSURE

Efficient performance monitoring for symmetric multi-threading systems is applicable to systems that have limited performance monitoring resources and enables efficient resource sharing on a per-execution unit basis. According to embodiments of the present invention, a special indicator may be allocated for an execution unit upon its request to start performance monitoring operation. The performance monitoring unit being shared is programmed to reset its counter and to start performance monitoring operation if there is only one execution unit requesting this operation. In case there are several requests pending, an attempt is made to program the performance monitoring unit to collect performance data for a subset of execution units the hardware is capable to support.

Upon a request to stop performance monitoring operation the previously allocated indicator may be removed, and the performance monitoring unit may be programmed to stop operating if there are no more active or pending requests. The final performance data may then be obtained. Otherwise, if performance monitoring was active for the current execution unit and there are some requests pending, the performance monitoring unit may be programmed to collect data for another execution unit. In case the performance monitoring was inactive for the current execution unit, this request may be discarded, and no performance data may be returned.

Thus, the sharing of performance monitoring unit according to embodiments of the present invention results in gathering of correct information pertaining to all execution units that requested the performance monitoring operation, the information being distributed over multiple execution units on an execution time basis, that is, the first or the last execution unit that becomes inactive may take all of the data previously.

1/3

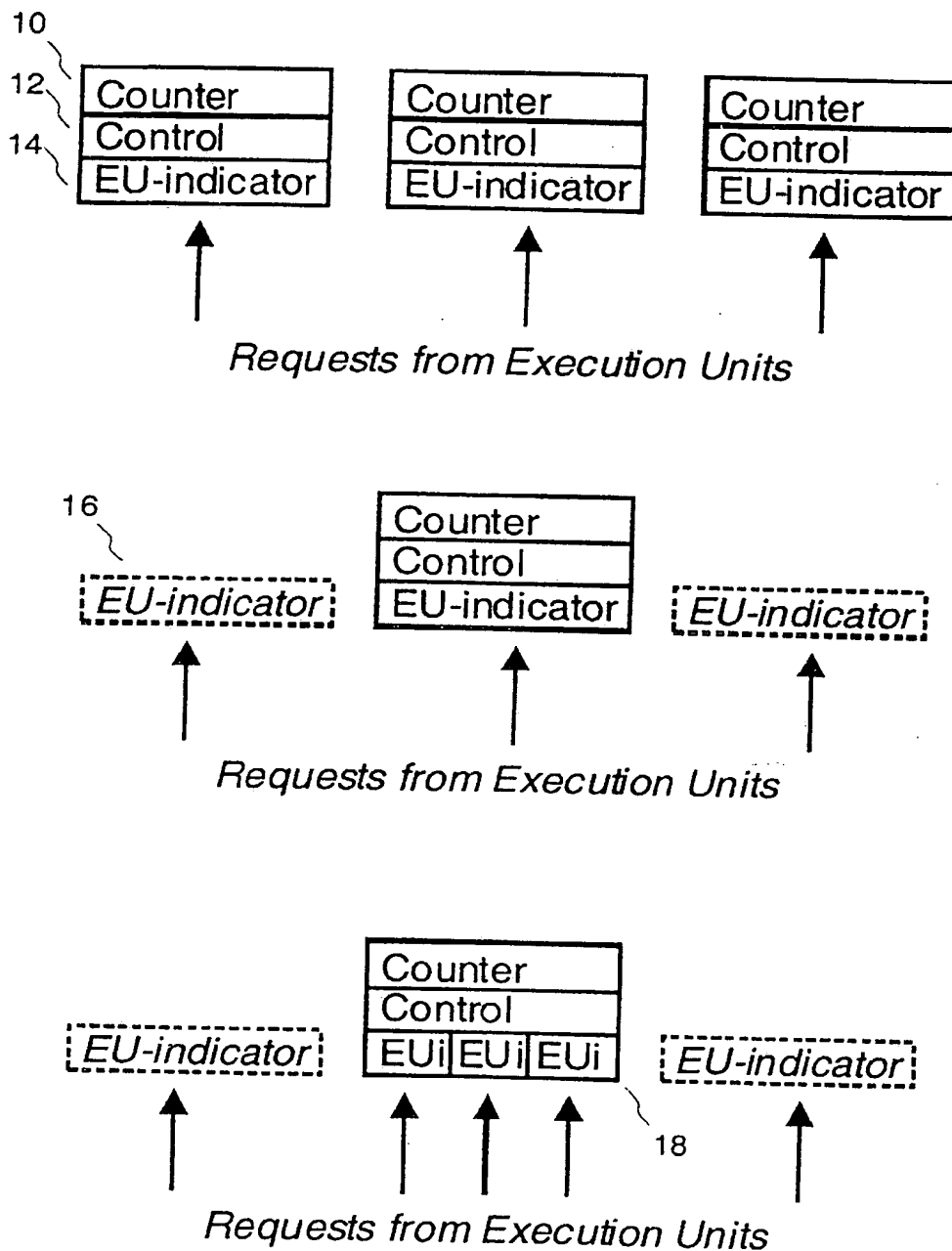


FIG. 1

2/3

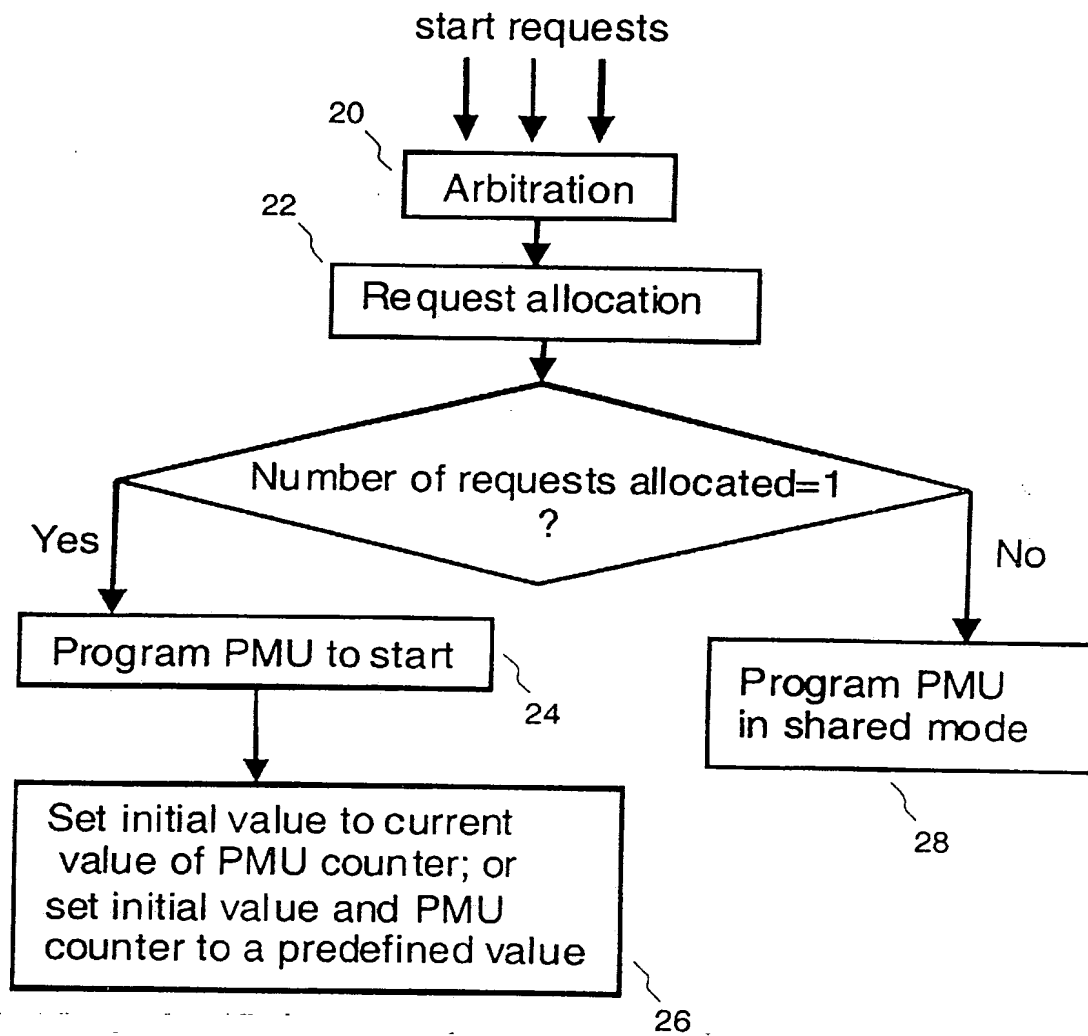


FIG. 2

3/3

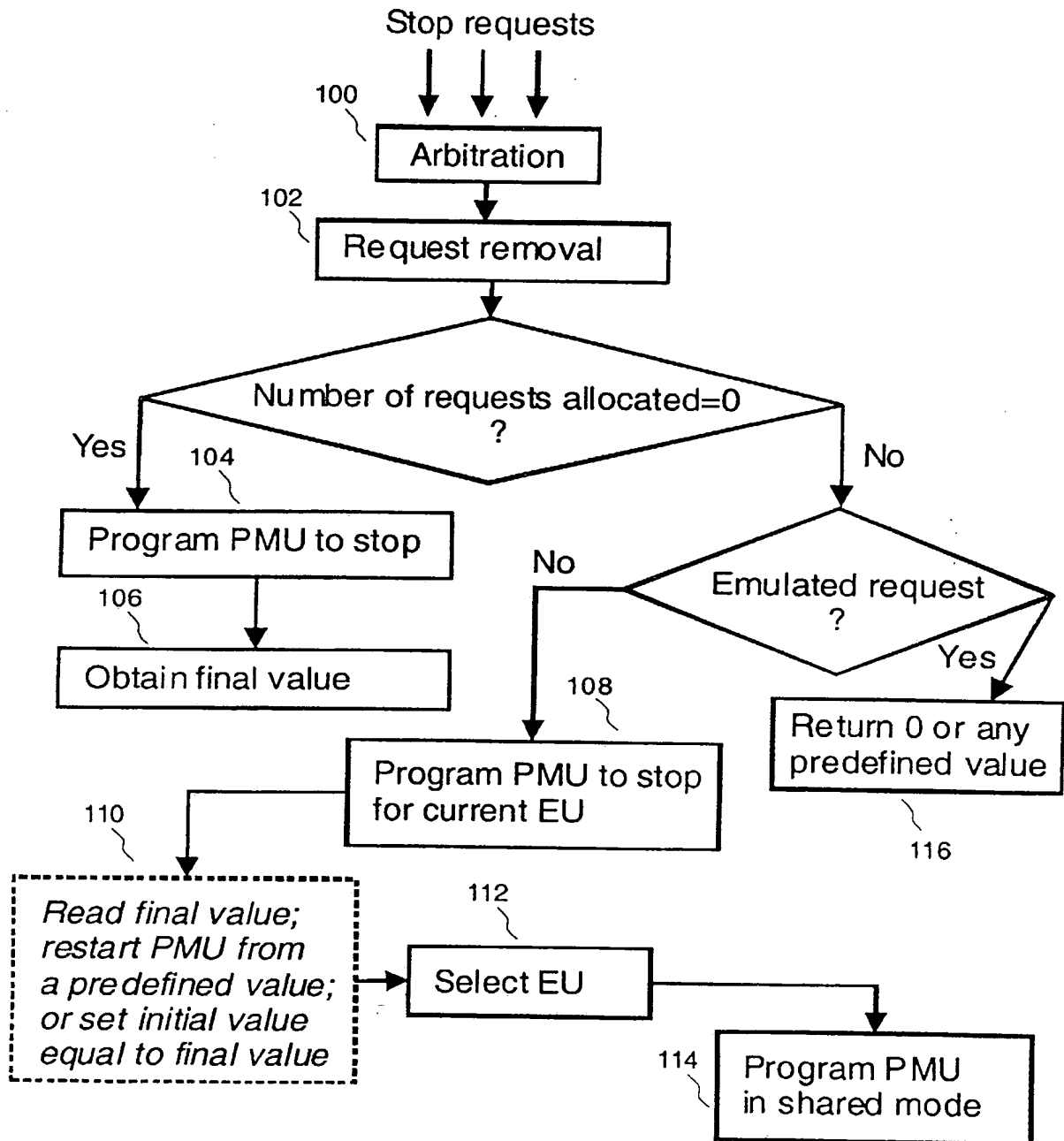


FIG. 3